IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of	COURTNEY	Confirmation No.:	9218
Serial No.	09/942,833	Examiner:	T. Q. Phan
Filed:	08/29/2001	Art Unit:	2128
FOR:	SYSTEM AND METHOD FOR MODELING A NETWORK DEVICE'S CONFIGURATION		

Mail Stop Appeal Briefs-Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

37 CFR 41.31 APPEAL BRIEF

Sir:

Applicant hereby appeals from the Final Action of December 12, 2005. The Notice of Appeal and a Pre-Appeal Brief Request for Review was filed on May 3, 2006.

REAL PARTY IN INTEREST

The real party in interest in this appeal is Intelliden Inc., as the assignee.

RELATED APPEALS AND INTERFERENCES

There are presently no related appeals or interferences.

STATUS OF CLAIMS

Claims 1 and 3-36 are pending, stand as rejected and are being appealed. Claims 1, 10, 19, 24 and 29 are independent. The appendix includes a true copy of all pending claims. No claims have been allowed.

STATUS OF AMENDMENTS

No amendments were filed subsequent to final rejection.

SUMMARY OF CLAIMED SUBJECT MATTER

The technology of the present invention relates generally to systems and methods for modeling configurations of network devices.

Although several embodiments of the present invention are disclosed in the specification, Figures 4, 5, 6 and the supporting text provides a good summary of embodiments which are exemplary of the subject matter defined by independent claims 1, 10, 19, 24 and 29. The main text describing Figures 4, 5, and 6 is located at paragraphs 30-37 of the specification, and paragraphs 10 and 11 help to clarify aspects of Figures 4, 5 and 6. Portions of these descriptions are reproduced or summarized below. Note that it is not Applicant's intention to limit the scope of the invention to what is described in this summary. This material is purely illustrative.

Figure 4, which is reproduced below for convenience, illustrates a system 220 that includes a document object model (DOM) generator 160 connected through a network 225 to a plurality of network devices 165, a system administrator 175, a schema storage device 170, and DOM applications 180.

The schema storage 170 includes a collection of schemas, each of which can include standard representations of the command structures for each of the network devices 165. For example, one schema could contain a representation of the command structure for all model 7500 Cisco™ routers using OS version 12.1, and another schema could contain a representation of the command structure routers using OS version 12.2.

In certain embodiments, these schemas can be directly used to generate an XML document that represents the configuration of a particular network device. In other embodiments, however, an intermediate representation (e.g., a hash representation) of the schema is generated and the intermediate representation is used to more quickly generate the corresponding XML document. By using the intermediate representation, the number of instruction cycles needed to generate the XML document is reduced significantly when compared to generating the XML document directly.

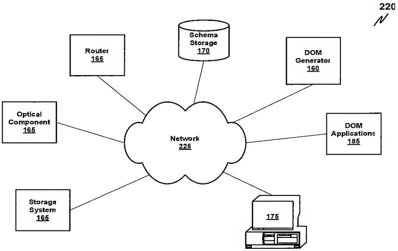

FIG 4

Figure 5, which is reproduced below depicts one implementation of the DOM generator 160 depicted in Figure 4. The DOM generator depicted in Figure 5 includes a schema hash system 230, an XML converter 235, and a DOM transformer 250. These components can be connected to the schema storage device 170, target network devices 165, a DOM storage device 245 and an XML storage device 250.

In this embodiment, the XML converter 235, using the appropriate schema, generates an XML document containing an XML representation of the network device's configuration. This XML document is then passed to the DOM transformer 240, which converts the XML document into a DOM. The output from the XML converter 235 and/or the DOM transformer 240 can be stored and passed to relevant software applications. For example, the output from the XML converter 235 can be stored in the XML storage device 250 and the output from the DOM transformer 240 can be stored in the DOM storage device 245.

Notably, the XML converter 235 of this embodiment can convert the native configuration of the network device 165 into an XML document using an intermediate representation of the schema associated with the network device 165, such as a hash table generated by the hash system 230, instead of the schema itself. By using an intermediate representation of the appropriate schema, the XML converter 235 can reduce the time and processing requirements needed to convert a native configuration into a corresponding XML document.

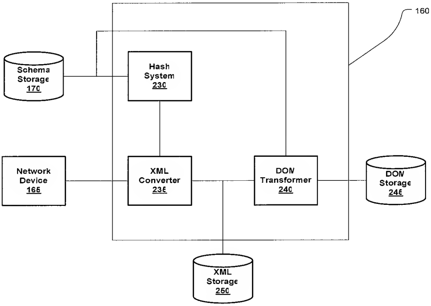

FIG 5

In operation, the DOM generator 160 determines the network device's characteristics by polling the network device or accessing a database (not shown) containing such information. Next, the XML converter 235 identifies the appropriate intermediate representation for the target network device 165. As previously described, this intermediate representation provides the necessary data to convert the native-format configuration of the target network device 165 into a standard format such as an XML format.

Possibly concurrently with the XML converter 235 identifying the corresponding intermediate representation, the XML converter 235 retrieves the configuration from the network device 165 and identifies each initial command within each configuration line. For example, the XML converter 235 could locate command distinguishing tags embedded in the configuration such as "begin command" and/or "end command." Alternatively, the XML converter 235 could use logical indicators within the configuration to distinguish the individual commands. Either way, using the identified initial command, the XML converter 235 generates a look-up key that is used to index the hash table, locate a hash map object that corresponds to the look-up key and retrieve that hash map object. The hash map object contains schema information regarding the command or value such as whether optional or required data type, etc.

Finally, using this hash map object, the XML converter 235 can assemble the XML-based command and write it to the corresponding XML document. The above process can be repeated for each command in the network device's native-format configuration to generate XML commands that may be assembled, as a model of the device configuration, into an XML document that can be stored in the XML storage device 250 and/or provided to the DOM transformer 240. Once the XML document has been assembled, it can be passed to the DOM transformer 240 where a DOM corresponding to the XML document can be generated.

GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

1) Whether claims 1 and 2-33 are rendered unpatentable under 35 U.S.C. 103(a) over U.S. Patent Application No. 2002/0191619 ("Shafer").

ARGUMENT

Claims 1 and 3-33 stand rejected under 35 U.S.C. § 103(a) as being unpatentable by Shafer. This rejection is improper because Shafer does not disclose at least a suggestion of each limitation of the claimed invention and the Final Action does not identify at least a suggestion of each limitation in the prior art. For efficiency, independent claims 1, 24 and 29 are initially addressed jointly and independent claims 10 and 19 are addressed independently.

For convenience, a summary of the Examiner's position relative to claims 1, 10, 19, 24 and 29, as well as the page number in this Appeal Brief where Applicant's remarks are found, is provided in the following tables:

Claim Limitations of claims 1, 24 and 29	**Examiner's Position re: Corresponding Structure in Shafer's Disclosure**	**Applicant's Response within this Appeal Brief**
determining a characteristic of the network device" that "comprises determining one of a network device manufacturer, network device model, and network device operating system version."	Nothing specifically identified, the Examiner merely cites paragraphs [0020], [0033], [0037], [0042], [0044], and [0057]-[0060]	Pages 9-11
retrieving a representation of a configuration schema…corresponding to the determined characteristic of the network device	Nothing specifically identified--Examiner cites paragraphs [0020], [0037], [0042], and [0044]	Pages 11-13
retrieving a first of the plurality of configuration commands	Nothing identified at all by the Examiner	Page 13
generating an XML object corresponding to the retrieved configuration command (Claim 1 only)	Nothing specifically identified--Examiner merely cites paragraphs [0033], [0035], [0037], [0042]-[0059]	Pages 13 and 14
generating a standard-format representation of the retrieved configuration command schema (Claims 24 and 29 only)	Nothing. Limitations completely ignored	Pages 14 and 15

Claim 10 Limitations	**Examiner's Position re: Corresponding Structure in Shafer's Disclosure**	**Applicant's Response within this Appeal Brief**
intermediate schema representation system (ISR)	The Examiner does not identify any structure with specificity. But indicates an "interface language for CLI" discloses the ISR and paragraphs: [0033]-[0038], [0042]-[0059], [0057]-[0060].	Page 16
XML converter configured to convert the native-format network device configuration into an XML document.	The Examiner admits Shafer does not teach an XML converter. The Examiner also mischaracterizes the claim limitations	Page 17
XML converter connected to the ISR	Nothing. The Examiner does not address these the connection between the XML converter and ISR	Page 18
a document object module (DOM) transformer connected to the XML converter, the DOM configured to transform the XML document into a DOM.	The Examiner does not identify any structure with specificity. But alleges Shafer teaches a "DOM implementation or generator available in several programming languages…" at paragraphs [0046]-[0050].	Pages 18 and 19

Claim 19 Limitations	**Examiner's Position re: Corresponding Structure in Shafer's Disclosure**	**Applicant's Response within this Appeal Brief**
a plurality of network devices;	Nothing. These limitations are completely ignored.	Pages 19 and 21
a DOM generator connected to the plurality of network devices	The Examiner does not identify any structure with specificity. But alleges [0046]-[0050] teach a DOM generator	Pages 20 and 21
a configuration schema storage device connected to the DOM generator	Nothing. These limitations are completely ignored.	Page 21
a DOM storage device connected to the DOM generator	Nothing specifically identified. The Examiner cites paragraphs [0046]-[0049],[0057], and[0060].	Pages 21 and 22

At the outset, Applicant would like to note that the Final Action provides so little detail to support each of its rejections that, in many instances, Applicant is left guessing as to what disclosures in Shafer allegedly correspond to the claimed limitations. More problematic, in many other instances, the Examiner completely ignores claim limitations altogether. As a consequence, each of the rejections are invalid on their face and the Examiner has not made a single prima facie rejection.

In addition, Shafer, the cited prior art, in many instances includes several different constructs within each of the paragraphs that the Examiner cites, yet the Examiner does not designate specific constructs within the cited paragraphs that allegedly correspond to the claim limitations; thus the Examiner has failed to honor Rule 37 CFR 1.104 (c)(2), which requires that, for references like Shafer, "the particular part relied on must be designated as nearly as practicable."

Independent claims 1, 24 and 29

Applicants submit that the 35 U.S.C. § 103(a) rejection against claims 1, 24 and 29 is improper because there are several limitations in these claims that are neither taught nor suggested by Shafer and the Final Action has not identified at least a suggestion of each claim limitation. Accordingly, the rejection against claims 1, 24 and 29 should be withdrawn. For simplicity, claim 1 is directly addressed, but unless indicated otherwise, the same arguments apply to claims 24 and 29.

Claim 1 is reproduced below for convenience.

> 1. A method for modeling a configuration corresponding to a network device, wherein the configuration includes a plurality of configuration commands, the method comprising:
>
> determining a characteristic of the network device, wherein determining the characteristic of the network device comprises determining one of a network device manufacturer, network device model, and network device operating system version;
>
> retrieving a representation of a configuration schema, the representation of a configuration schema corresponding to the determined characteristic of the network device;
>
> retrieving a first of the plurality of configuration commands from the network device configuration corresponding to the network device; and
> generating an XML object corresponding to the retrieved configuration command;
>
> wherein the XML object is generated according to at least a portion of the retrieved representation of the configuration schema.

Shafer neither discloses nor suggests "determining one of a network device manufacturer, network device model, and network device operating system"

As shown, claim 1 recites, "determining a characteristic of the network device" that "comprises determining one of a network device manufacturer, network device model, and network device operating system version."

Shafer neither discloses nor suggests "determining one of a network device manufacturer, network device model, and network device operating system." For example, a simple word search on the Shafer patent reveals that it does not once mention manufacture, network device model or network device operating system version. Nor does Shafer suggest that that is advantageous to determine a network manufacture, network device model or network device operating system version.

In contrast to Applicant's system and method for modeling configurations of network devices (e.g., the network devices 165 depicted in FIG. 4) where a network device manufacture, network device model or network device operating system version can be utilized to retrieve a configuration schema (e.g., from schema storage 170 depicted in FIG. 4) that corresponds to a target network device, Shafer is directed to an application programming interface (API) for their router 10 (See Shafer, Abstract, Para. 0031), and neither Shafer nor the Final Action articulate any reason why it would be advantageous for Shafer to determine a network device manufacture, network device model or network device operating system version.

The Examiner ignores the "determining one of a network device manufacturer, network device model, and network device operating system" limitations and provides no support for an allegation that Shafer teaches these limitations.

In rejecting claim 1, the Final Action mimics back the language of the claim but never specifically points out where Shafer determines a manufacture, network device model or network device operating system version. Instead, the Final Action ignores these limitations and alleges (on pages 3 and 11) that Shafer discloses several items that neither appear in claim 1 nor Shafer. In particular, the Examiner alleges that paragraphs [0020], [0033], [0037], [0042], [0044], and [0057]-[0060], disclose "[d]etermining characteristics of the network device for interfacing (such as power, voltage, current, ports, i/o bandwidth, model device type, device configuration file for operation, etc)." Again, these listed features do not appear in claim 1 so their relevance is unclear. But it should be noted that a simple review of the cited paragraphs reveals that the Examiner is clearly mistaken: Shafer does not even hint at determining power, voltage, current, ports, etc.

The Final Action does allege that Shafer teaches determining "model device type," but this collection of words is not found in claim 1, so Applicants are unsure whether the Final Action is referring to the recited "network device model." But assuming, arguendo, that the Examiner is alleging a "model device type" corresponds to the claimed "network device model," the Examiner does not identify with any specificity where Shafer allegedly teaches determining a "model device type." Specifically, Applicant has searched paragraphs [0020], [0033], [0037], [0042], [0044] and [0057]-[0060] and are unable to find a single recitation of "model device type," "model device," "device type," "model type," "model," or even "type," and the Final Action does not identify with any specificity where in Shafer a "model device type" is determined.

In addition, Applicants have reviewed both Shafer as a whole and paragraphs [0020], [0033], [0037], [0042], [0044] and [0057]-[0060] and neither "manufacture," "operating system version," nor "model" is even found. Although Shafer does teach that their router 10 includes a routing engine 14 with an operating system 24 as shown in FIG. 2, Shafer does not teach that a version of their operating system is determined for any reason.

Shafer neither discloses nor suggests "retrieving a representation of a configuration schema...corresponding to the determined characteristic of the network device"

Claim 1 also recites "retrieving a representation of a configuration schema...corresponding to the determined characteristic of the network device." Shafer, however, does not suggest retrieving "a representation of a configuration schema." And as a consequence, Shafer certainly does not teach retrieving a configuration schema "corresponding to the determined characteristic of the network device." Although the

Final Action enumerates paragraphs [0020], [0037], [0042], and [0044] of Shafer, these paragraphs disclose a lot of subject matter unrelated to claim 1; yet the Final Action does not identify any specific language or construct within Shafer that allegedly suggests retrieving a representation of a configuration schema; thus Applicants are left to guess about what the Examiner is referring to, and Applicant's best guess is that the Examiner is alleging that Shafer's management interface schema 54 corresponds to the claimed "configuration schema."

As discussed above, Applicant's configuration schemas can include standard representations of the command structures for each of the network devices 165. In contrast, Shafer's management interface schema 54

> maps extensible markup language tags received by management server module 32 to information associated with software modules 48, 50, including the information in database 52 and information that may be obtained directly from software modules 48, 50.

(Shafer, Para. [0042]).

Thus Shafer's management interface schema 54 is very different from the recited "configuration schema." Moreover, Shafer does not teach that their management interface schema 54 is "retrieved" at all—instead it is used to map extensible markup language tags to information associated with their software modules 48, 50. As a consequence, Shafer's management interface schema 54 can not correspond to the claimed "configuration schema."

The Final Action ignores the connection between the "determined characteristic" and the retrieved "representation of a configuration schema."

As recited in claim 1, the retrieved representation of the configuration schema corresponds to the determined characteristic of the network device. As discussed, Shafer does not even retrieve their management interface schema 54, as a consequence, Shafer certainly does not retrieve a configuration schema corresponding to a determined characteristic of a network device. In addition, the Final Action makes no connection between Shafer's management interface schema 54 and any alleged determined characteristic of a network device. Accordingly, the rejection against claim 1 cannot properly stand.

Shafer neither discloses nor suggests "retrieving a first of the plurality of configuration commands"

Claim 1 also recites "retrieving a first of the plurality of configuration commands." Shafer, however, neither teaches nor suggests retrieving a first plurality of configuration commands. Moreover, the final action *does not even allege* that Shafer teaches or suggests retrieving a first of the plurality of configuration commands. Instead, the final action merely reproduces the claim language, but the Examiner does not even attempt to identify a single paragraph in Shafer that allegedly teaches retrieving configuration commands. Thus the rejection is improper on its face.

Shafer neither discloses nor suggests "generating an XML object corresponding to the retrieved configuration command" as recited in claim 1.

With respect only to claim 1 (i.e., this argument does not apply to claims 24 and 29), claim 1 recites "generating an XML object corresponding to the retrieved configuration command" but again, the Final Action provides no specificity as to what construct within the teachings of Shafer correspond to the recited "XML object." Instead, the Final Action merely lists paragraphs [0033], [0035], [0037], [0042]-[0059] of Shafer. Although Applicant is again left guessing, presumably the Examiner contends that Shafer's XML API 62 corresponds to the recited "XML object." Assuming that this is the Examiner's position, Shafer's XML API 62 can not correspond to the recited "XML object" because Shafer's XML API 62 is not generated according to at least a portion of a representation of a configuration schema. Nor does Shafer's XML API 62 correspond to a retrieved configuration command.

As discussed, the Examiner has not even specified what allegedly corresponds to the claimed representation of a configuration schema, but assuming the Examiner contends that Shafer's management interface schema 54 corresponds to Applicant's representation of a configuration schema, Shafer's XML API 62 is not generated according to their management interface schema 54—instead their management interface schema 54 is used to map extensible markup language tags to information associated with their software modules 48, 50. Moreover, Shafer neither teaches nor suggests that their XML API 62 corresponds to a retrieved command. Thus, Shafer's XML API 62 can not correspond to the claimed XML object.

Shafer neither discloses nor suggests "generating a standard-format representation of the retrieved configuration command...according to at least a portion of the retrieved representation of the configuration schema" as recited in claims 24 and 29.

With respect only to claims 24 and 29, both of these claims recite "generating a standard-format representation of the retrieved configuration command...according to at least a portion of the retrieved representation of the configuration schema." As discussed, Shafer does not teach retrieving a configuration command. Moreover, Shafer does not teach generating a standard-format representation of the retrieved configuration command—there is simply no teaching or suggestion of such in Shafer. And as a consequence, Shafer certainly does not teach generating a standard-format representation of the retrieved configuration command according to at least a portion of the retrieved representation of the configuration schema.

Moreover, the Final Action does not even allege that Shafer teaches these limitations. In particular on pages 7 and 9 where the Examiner addresses claims 24 and 29, the "standard-format representation" language does not even appear. Thus the rejection is improper on its face.

In short, claims 1, 24 and 29 include several limitations neither taught nor suggested by the prior art. Moreover, the rejection of claims 1, 24 and 29 is improper on its face because the Final Action does not identify at least a suggestion of each limitation in the prior art and in some instances the Final Action does not even allege that some limitations are found in the prior art. Accordingly, the rejection against claims 1, 24 and 29 and the corresponding dependent claims can not properly stand.

Independent Claim 10

Applicants submit that the 35 U.S.C. § 103(a) rejection against claim 10 is improper because Shafer neither teaches nor suggests many limitations of claim 10 and the Final Action does not identify at least a suggestion of each limitation in the prior art. Accordingly, Applicants submit that the rejection against claim 10 and the corresponding dependent claims should be withdrawn.

Claim 10 recites a specific architecture for modeling a native-format network device configuration. That architecture includes several components nether taught nor suggested by Shafer, including an intermediate schema representation system (ISR), an XML converter connected to the ISR, and a document object model (DOM) transformer connected to the XML converter. Claim 10 is reproduced below.

> 10. A system for modeling a native-format network device configuration, the system comprising:
> an intermediate schema representation system (ISR);
> an XML converter connected to the ISR, the XML converter configured to convert the native-format network device configuration into an XML document; and
> a document object model (DOM) transformer connected to the XML converter, the DOM transformer configured to transform the XML document into a DOM.

Shafer does not teach or suggest an "intermediate schema representation system," and the Final Action does not identify--with any specificity--any construct within Shafer that allegedly corresponds to the recited "intermediate schema representation system (ISR)"

Instead of identifying limitations recited in claim 10, the Final Action appears to recite limitations from claim 1 (See Final Action pages 4 and 5).

The Examiner, at page 11, however, does state that Shafer discloses an "interface language for CLI (it's called intermediate schema representation) for system interfacing." The rejection is not clearly expressed, so Applicant is unsure what the Examiner is talking about. But, it is clear that the Examiner does *not* indicate how such an "interface language" can correspond to the recited intermediate schema representation system (ISR). Nor does the Final Action provide any specificity as what constructs in Shafer allegedly corresponds to the recited "intermediate schema representation system." As a consequence the rejection is improper under both 37 CFR 1.104 (c)(2) and §103(a).

Shafer neither teaches nor suggests an "XML converter configured to convert the native-format network device configuration into an XML document," and the basis presented in the Final Action, for alleging that Shafer suggests these limitations, is for the most part, unintelligible and mischaracterizes Shafer and the claim limitations.

The Examiner admits that Shafer does not teach an XML converter, but the Examiner ignores the claim language and instead states:

> [p]ractitioner in the art at the time of the invention was made would have found Shafer disclosure in the present US application with feature limitations of network router interface, command formats translation, router configuration changes, etc would require the claimed limitation of the XML converter for converting or translating CLI into XML format as claimed.

(Final Action, page 5).

Applicant is frankly confused by this statement and uncertain about what point the Examiner is making. If the Examiner is setting forth a motivation to alter Shafer to include an XML converter, the motivation the Examiner expresses is lost in the incoherent mix of Shafer's features and the recitation of "limitations" not found in claim 10. For example, Applicant is uncertain whether the Examiner is discussing claim 10, Shafer, or both, and if both, it is unclear when Shafer is being characterized and when claim 10 is being discussed. But in any case, the Examiner has mischaracterized claim 10, Shafer or both.

Shafer neither teaches nor suggests that their router translates any commands from one format to another. Instead, Shafer's XML API 62 merely provides Shafer's clients 56, 60 with an XML-based API in response to a particular command. In particular, as described by Shafer:

> In accordance with the principles of the invention, however, the command line interface presented by control unit 12 is dynamically replaced with an XML-based API upon receipt of a particular CLI command from a client. More specifically, upon receipt of the command, referred to herein as the "xml-mode" command, management server module 32 receives subsequent incoming commands directly and, as described below, services the XML encoded CLI commands based on the XML API.

(Shafer, Para. [0038]; See also, Para. [0041]).

Thus, instead of translating any commands--as the Examiner contends--Shafer merely receives XML encoded commands in response to a particular CLI command.

Moreover, contrary to the Examiner's statement, claim 10 does not include any limitations that require the claimed XML converter to carry out "converting or translating CLI into XML." Nor does claim 10 include any limitations directed to a "network router interface."

In addition, the Examiner ignores that the claimed XML converter is "connected to the ISR." There is simply no attempt in the Final Action to show two constructs that are both connected and that correspond to the recited ISR and the XML converter.

Moreover, the Final Action ignores that the recited XML converter is "configured to convert the native-format network device configuration into an XML document." Specifically, the Final Action does not even allege that Shafer teaches or suggests any construct that converts a network device configuration into an XML document. And a review of Shafer reveals that Shafer simply does not suggest anything that converts a network device configuration into an XML document. In particular, Shafer does not suggest a conversion of any network device configurations, nor does Shafer suggest converting anything into an XML document. Instead Shafer teaches an XML API 62 that enables their clients to communicate with their router 10 via the XML API 62.

The Final Action fails to even allege that Shafer teaches or suggests a "document object module (DOM) transformer connected to the XML converter, the DOM configured to transform the XML document into a DOM."

The Final Action, at page 11, contends that Shafer teaches a "DOM implementation or generator available in several programming languages...." But the Final Action does not even allege that Shafer teaches or suggests the recited "document object model (DOM) transformer." Instead the Final Action cites language not found in claim 10. Nor does the Final Action identify, with any specificity what construct within Shafer allegedly corresponds to the DOM transformer. As a consequence the rejection is improper under both 37 CFR 1.104 (c)(2) and §103(a).

The material cited in the Final Action does not, at least, suggest each and every limitation in claim 10. Accordingly, applicants submit that the 103(a) rejection against claim 10 is improper.

Independent claim 19

Applicant submits that the 35 U.S.C. §103(a) rejection against claim 19 is improper because Shafer does not at least suggest each claimed limitation of claim 19 and the Final Action completely ignores several limitations of claim 19. Accordingly, Applicant submits that the rejection against claim 19 and the corresponding dependent claims should be withdrawn.

Claim 19 is reproduced below.

> 19. A system for modeling a network device configuration, the system comprising:
> a plurality of network devices;
> a DOM generator connected to the plurality of network devices;
> a configuration schema storage device connected to the DOM generator; and
> a DOM storage device connected to the DOM generator.

Shafer neither teaches nor suggests a DOM generator connected to a plurality of network devices.

Again, Shafer teaches an XML API 62 for their router 10 that enables their clients 56. 60 to communicate with their router 10 via the XML API 62—there is simply nothing in the teaching of Shafer that suggests a DOM generator connected to a plurality of network devices.

The Final Action does not identify with any specificity any construct that allegedly corresponds to the recited "DOM generator."

In addition, the Final Action does not identify with any specificity any construct that allegedly corresponds to the recited "DOM generator" that is connected to the plurality of network devices. Instead the Final Action alleges, without pointing to anything specifically, that paragraphs [0046]-[0050] teach a DOM generator. A word search of Shafer, however, reveals that Shafer does not disclose a "DOM generator" and the Final Action does not identify any construct that allegedly corresponds to the recited DOM generator; thus the rejection is improper under both 37 CFR 1.104 (c)(2) and §103(a). Assuming, arguendo, that Shafer somewhere discloses a DOM generator, claim 19 requires that the DOM generator be connected to both a plurality of network devices and a configuration schema storage device. Shafer does not suggest a DOM generator coupled to a plurality of network devices and Shafer certainly does not teach DOM generator that is connected to both a plurality of network devices and a configuration schema storage device.

The Final Action completely ignores several limitations of claim 19.

In addition, the Final Action again completely ignores several claim limitations. Specifically, instead of attempting to identify at least a suggestion of the limitations of claim 19 within Shafer, the Final Action appears to be reciting many limitations of claim 1—which do not appear in claim 19 (See Final Action, page 6).

The Final Action ignores the recited "plurality of network devices."

The Final Action does not even make an attempt to identify a suggestion of the recited "plurality of network devices." In particular, Applicants have been unable to find the recited "plurality of network devices" even mentioned in the Final Action; thus the rejection is clearly improper under both 37 CFR 1.104 (c)(2) and §103(a).

Shafer does not suggest, and the Final Action ignores, the recited connection between the DOM generator and the plurality of network devices.

The DOM generator is connected to the plurality of network devices; yet the Final Action makes no mention of this connection.

Shafer does not suggest, and the Final Action ignores, the "configuration schema storage device connected to the DOM generator" that is recited in claim 19.

Again, a proper rejection requires that at least a suggestion of each limitation be identified in the prior art. The Final Action, however, does not even mention the "configuration schema storage device;" thus the rejection is clearly improper under both 37 CFR 1.104 (c)(2) and §103(a).

Although Shafer teaches a management interface schema 54, as discussed above, Shafer's management interface schema 54 merely maps "extensible markup language tags received by management server module 32 to information associated with software modules 48, 50"-- it is not a configuration schema storage device. Nor is Shafer's management interface schema 54 coupled to a DOM generator as required by claim 19.

Shafer does not teach a DOM storage device and the Final Action ignores the connection between the DOM storage device and the DOM generator recited in claim 19.

Finally, claim 19 recites a "DOM storage device connected to the DOM generator." Shafer neither teaches nor suggests a DOM storage device. In addition, the

Final Action ignores these limitations and instead alleges that Shafer discloses "[m]eans for storing DOM and means for temporarily storing generated DOM implementation ([0046]-[0049],[0057],[0060])." Although Applicant is again left to guess, presumably the Final Action contends that Shafer teaches a "means for storing DOM," and the "means for storing DOM" corresponds to the claimed "DOM storage device." Assuming, arguendo, that somewhere, Shafer does teach a "means for storing DOM," the Final Action merely points to paragraphs [0046]-[0049],[0057],[0060] without identifying--with any specificity--what construct within these portions of Shafer allegedly discloses "means for storing DOM." As a consequence, the rejection is improper under both 37 CFR 1.104 (c)(2) and §103(a). Moreover, in claim 19, the recited "DOM storage device" is connected to "the DOM generator." The Final Action does not attempt to identify where the alleged "means for storing DOM" is connected to a "DOM generator," and as a consequence, the rejection of claim 19 is also improper for this additional reason.

The material cited in the Final Action does not provide at least a suggestion of each limitation of claim 19. Accordingly, the rejection against claim 19 and corresponding dependent claims is improper.

SUMMARY

All of the pending claims are patentable for the reasons set forth herein, and Appellant respectfully requests such finding.

Three copies of this Appeal Brief are provided along with payment of the required fee.

COOLEY GODWARD LLP
ATTN: Patent Group
The Bowen Building
875 15th Street NW, Ste. 800
Washington, DC 20005-2221
Tel: (720) 566-4035
Fax: (202) 842-7899

Respectfully submitted,
COOLEY GODWARD LLP

By: ______________________
Sean R. O'Dowd
Reg. No. 53,403

CLAIMS APPENDIX

1. A method for modeling a configuration corresponding to a network device, wherein the configuration includes a plurality of configuration commands, the method comprising:

determining a characteristic of the network device, wherein determining the characteristic of the network device comprises determining one of a network device manufacturer, network device model, and network device operating system version;

retrieving a representation of a configuration schema, the representation of a configuration schema corresponding to the determined characteristic of the network device;

retrieving a first of the plurality of configuration commands from the network device configuration corresponding to the network device; and

generating an XML object corresponding to the retrieved configuration command; wherein the XML object is generated according to at least a portion of the retrieved representation of the configuration schema.

2. (cancelled)

3. The method of claim 1, wherein the representation of the configuration schema comprises a plurality of schema portions and wherein retrieving the representation of the configuration schema comprises:

retrieving an intermediate representation of the configuration schema, wherein the intermediate representation comprises a plurality of keys;

wherein each of the plurality of keys is associated with a corresponding one of the plurality of schema portions.

4. The method of claim 3, wherein retrieving the intermediate representation of the configuration schema comprises:

retrieving a hash table.

5. The method of claim 3, further comprising:

generating a look-up key for the retrieved configuration command.

6. The method of claim 5, further comprising:

identifying a first of the plurality of keys in the intermediate representation, the first of the plurality of keys corresponding to the generated look-up key; and

retrieving a first of the plurality of schema portions, the first of the plurality of schema portions corresponding to the first of the plurality of keys;

wherein the XML object is generated according to the first of the plurality of schema portions.

7. The method of claim 1, further comprising:

converting the XML object to an XML document.

8. The method of claim 7, further comprising:

converting the XML document into a document object model (DOM).

9. The method of claim 8, further comprising:

verifying the DOM against the at least the representation of the configuration schema.

10. A system for modeling a native-format network device configuration, the system comprising:

an intermediate schema representation system (ISR);

an XML converter connected to the ISR, the XML converter configured to convert the native-format network device configuration into an XML document; and

a document object model (DOM) transformer connected to the XML converter, the DOM transformer configured to transform the XML document into a DOM.

11. The system of claim 10, wherein the native-format network device configuration is associated with a router.

12. The system of claim 10, wherein the native-format network device configuration is associated with a data storage system.

13. The system of claim 10, wherein the native-format network device configuration is associated with an optical component.

14. The system of claim 10, further comprising:

a DOM storage device for storing the DOM.

15. The system of claim 14, wherein the DOM storage device comprises temporary storage.

16. The system of claim 14, further comprising:

an XML-to-XML converter connected to the DOM storage device.

17. The system of claim 14, further comprising:

an XML-to-CLI converter connected to the DOM storage device.

18. The system of claim 14, further comprising:

a graphical user interface connected to the DOM storage device.

19. A system for modeling a network device configuration, the system comprising:

a plurality of network devices;

a DOM generator connected to the plurality of network devices;

a configuration schema storage device connected to the DOM generator; and

a DOM storage device connected to the DOM generator.

20. The system of claim 19, further comprising:

a DOM application connected to the DOM generator.

21. The system of claim 19, wherein the configuration schema storage device comprises:

an intermediate schema representation storage device.

22. The system of claim 19, further comprising:

an XML-to-XML converter connected to the DOM generator.

23. The system of claim 19, further comprising:

an XML-to-CLI converter connected to the DOM generator.

24. A method for modeling a configuration corresponding to a network device, wherein the configuration includes a plurality of configuration commands, the method comprising:

determining a characteristic of the network device, wherein determining the characteristic of the network device comprises determining one of a network device manufacturer, network device model, and network device operating system version;

retrieving a representation of a configuration schema, the representation of a configuration schema corresponding to the determined characteristic of the network device;

retrieving a first of the plurality of configuration commands from the network device configuration corresponding to the network device; and

generating a standard-format representation of the retrieved configuration command;

wherein the standard-format representation is generated according to at least a portion of the retrieved representation of the configuration schema.

25. The method of claim 24, wherein the representation of the configuration schema comprises a plurality of schema portions and wherein retrieving the representation of the configuration schema comprises:

retrieving an intermediate representation of the configuration schema, wherein the intermediate representation comprises a plurality of keys;

wherein each of the plurality of keys is associated with a corresponding one of the plurality of schema portions.

26. The method of claim 25, further comprising:

generating a look-up key for the retrieved configuration command.

27. The method of claim 26, further comprising:

identifying a first of the plurality of keys in the intermediate representation, the first of the plurality of keys corresponding to the generated look-up key; and

retrieving a first of the plurality of schema portions, the first of the plurality of schema portions corresponding to the first of the plurality of keys;

wherein the standard-format representation is generated according to the first of the plurality of schema portions.

28. The method of claim 24, wherein the standard-format representation comprises an XML object.

29. A system for modeling a configuration corresponding to a network device, wherein the configuration includes a plurality of configuration commands, the system comprising:

a processor;

a storage device connected to the processor; and

a plurality of instructions stored on the storage device, the plurality of instructions configured to cause the processor to:

determine a characteristic of the network device, wherein the instructions configured to determine the characteristic of the network device comprise instructions configured to determine one of a network device manufacturer, network device model, and network device operating system version;

retrieve representation of a configuration schema, the representation of a configuration schema corresponding to the determined characteristic of the network device;

retrieve a first of the plurality of configuration commands from the network device configuration corresponding to the network device; and

generate a standard-format representation of the retrieved configuration command;

wherein the standard-format representation is generated according to at least a portion of the retrieved representation of the configuration schema.

30. The system of claim 29, wherein the representation of the configuration schema comprises a plurality of schema portions and wherein the plurality of instructions cause the processor to retrieve the at least the representation of the configuration schema by:

retrieving an intermediate representation of the configuration schema, wherein the intermediate representation comprises a plurality of keys;

wherein each of the plurality of keys is associated with a corresponding one of the plurality of schema portions.

31. The system of claim 29, wherein the plurality of instructions are further configured to cause the processor to:

generate a look-up key for the retrieved configuration command.

32. The system of claim 31, wherein the plurality of instructions are further configured to cause the processor to:

identify a first of the plurality of keys in the intermediate representation, the first of the plurality of keys corresponding to the generated look-up key; and

retrieve a first of the plurality of schema portions, the first of the plurality of schema portions corresponding to the first of the plurality of keys;

wherein the standard-format representation is generated according to the first of the plurality of schema portions.

33. The system of claim 29, wherein the standard-format representation comprises an XML object.

34. The system of claim 31, wherein the plurality of instructions are further configured to cause the processor to:

convert the XML object to an XML document.

35. The system of claim 34, wherein the plurality of instructions are further configured to cause the processor to:

convert the XML document into a document object model (DOM).

36. The system of claim 35, wherein the plurality of instructions are further configured to cause the processor to:

verify the DOM against the at least the representation of the configuration schema.

EVIDENCE APPENDIX

None

RELATED PROCEEDINGS APPENDIX

None